

FILEID**LWRITE

M 16

LL	WW	WW	RRRRRRRR		TTTTTTTTTT	EEEEEEEEE
LL	WW	WW	RRRRRRRR		TTTTTTTTTT	EEEEEEEEE
LL	WW	WW	RR RR		TT	EE
LL	WW	WW	RR RR		TT	EE
LL	WW	WW	RR RR		TT	EE
LL	WW	WW	RRRRRRRR		TT	EE
LL	WW	WW	RRRRRRRR		TT	EE
LL	WW	WW	RR RR		TT	EE
LL	WW	WW	RR RR		TT	EE
LL	WWWW	WWWW	RR RR		TT	EE
LL	WWWW	WWWW	RR RR		TT	EE
LLLLLLLL	WW	WW	RR RR		TT	EEEEEEEEE
LLLLLLLL	WW	WW	RR RR		TT	EEEEEEEEE

LL		SSSSSSSS
LL		SSSSSSSS
LL		SS
LLLLLLLL		SSSSSSSS
LLLLLLLL		SSSSSSSS

```
0001 0 ZTITLE 'EDTSLWRITE - write to a file'  
0002 0 MODULE EDTSLWRITE {  
0003 0 IDENT = 'V04-000'  
0004 0 )=  
0005 1 BEGIN  
0006 1  
0007 1 *****  
0008 1 *  
0009 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY  
0010 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.  
0011 1 * ALL RIGHTS RESERVED.  
0012 1 *  
0013 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
0014 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
0015 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
0016 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
0017 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
0018 1 * TRANSFERRED.  
0019 1 *  
0020 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
0021 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
0022 1 * CORPORATION.  
0023 1 *  
0024 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
0025 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.  
0026 1 *  
0027 1 *  
0028 1 *****  
0029 1  
0030 1  
0031 1 **  
0032 1 FACILITY: EDT -- The DEC Standard Editor  
0033 1  
0034 1 ABSTRACT:  
0035 1  
0036 1 This module executes the line mode EXIT and WRITE commands.  
0037 1  
0038 1 ENVIRONMENT: Runs at any access mode - AST reentrant  
0039 1  
0040 1 AUTHOR: Bob Kushlis, CREATION DATE: February 3, 1978  
0041 1  
0042 1 MODIFIED BY:  
0043 1  
0044 1 1-001 - Original. DJS 30-JAN-1981. This module was created by  
0045 1 extracting the routines WRITE_FILE, EDT$SEX1_CMD , and EDTSSWR_CMD  
0046 1 from the module EXEC.BLI.  
0047 1 1-002 - Regularize headers. JBS 23-Mar-1981  
0048 1 1-003 - Change WRITE_COM to EDTSSWR_CMD . JBS 30-Mar-1981  
0049 1 1-004 - Use the new message codes. JBS 04-Aug-1981  
0050 1 1-005 - Don't write out summary when flag clear. STS 05-Oct-1981  
0051 1 1-006 - Convert open output file to use EDT$FILEIO. STS 13-Jan-1982  
0052 1 1-007 - Remove division from line number calculations. SMB 15-Jan-1982  
0053 1 1-008 - Convert reads and writes to use EDT$FILEIO. STS 15-Jan-1982  
0054 1 1-009 - Add capability for 15 digit line numbers. SMB 19-Jan-1982  
0055 1 1-010 - pass sequence numbers by descriptors. STS 20-Jan-1981  
0056 1 1-011 - Change the checking of sequence/nosequence. STS 21-Jan-1982  
0057 1 1-012 - Fix size bugs related to 15 digit sequences & increments. SMB 26-Jan-1982
```

58 0058 1 | 1-013 - Remove original line numbers. SMB 28-Jan-1982
 59 0059 1 | 1-014 - Fix bug in writing with /SEQ. SMB 30-Jan-1982
 60 0060 1 | 1-015 - Add range checking on seq.# and incr. SMB 5-Feb-1982
 61 0061 1 | 1-016 - Take out call to edt\$\$_get_fnam. STS 10-Feb-1982
 62 0062 1 | 1-017 - Change line number division to a routine call. SMB 11-Feb-1982
 63 0063 1 | 1-018 - Format filename for output. STS 12-Feb-1982
 64 0064 1 | 1-019 - Pass filename to edt\$\$fiopn err. STS 26-Feb-1982
 65 0065 1 | 1-020 - Add literals for callable EDT. STS 08-Mar-1982
 66 0066 1 | 1-021 - Use the input file name as the output name, if permitted and
 there is no output file name. JBS 25-Mar-1982
 67 0067 1 | 1-022 - The file I/O routines return a status. JBS 26-Mar-1982
 68 0068 1 | 1-023 - Correct a typo in edit 1-022. JBS 27-Mar-1982
 69 0069 1 | 1-024 - Minor edits, plus remove code that doesn't seem to be executed. SMB 30-Mar-1982
 70 0070 1 | 1-025 - Rearrange where file name is stored so we get "real" name instead
 of "temp" filename. SMB 31-Mar-1982
 71 0071 1 | 1-026 - Add code for CONTROL C check during record writing. SMB 08-Apr-1982
 72 0072 1 | 1-027 - Print messages on CLOSE errors. JBS 12-Apr-1982
 73 0073 1 | 1-028 - Convert PDP-11 command file names to uppercase. SMB 13-Apr-1982
 74 0074 1 | 1-029 - Clear PREV_RANGE when creating a range block for EXIT. JBS 15-Apr-1982
 75 0075 1 | 1-030 - Don't set up any message for nosummary. STS 16-Apr-1982
 76 0076 1 | 1-031 - Change NO FILE error message and processing for WRITE. SMB 21-Apr-1982
 77 0077 1 | 1-032 - Save original buffer address. STS 10-May-1982
 78 0078 1 | 1-033 - Make minor modifications based on code review input. SMB 24-May-1982
 79 0079 1 | 1-034 - Set a flag if control C actually aborts something. JBS 24-May-1982
 80 0080 1 | 1-035 - Fix bug with explicit file names. SMB 25-May-1982
 81 0081 1 | 1-036 - Pass default file name using RHB parameter. JBS 15-Jun-1982
 82 0082 1 | 1-037 - Stop working message before printing messages to screen. SMB 22-Jun-1982
 83 0083 1 | 1-038 - Stop processing if bad select range. SMB 01-Jul-1982
 84 0084 1 | 1-039 - Use EDT\$\$FMT_CRLF instead of EDT\$\$OUT_FMTBUF. JBS 05-Jul-1982
 85 0085 1 | 1-040 - Change write file messages. SMB 13-Jul-1982
 86 0086 1 | 1-041 - Give message for write errors. STS 21-Jul-1982
 87 0087 1 | 1-042 - Don't put out error messages on 11's STS 22-Jul-1982
 88 0088 1 | 1-043 - Restore screen width before displaying file name on EXIT. SMB 29-Jul-1982
 89 0089 1 | 1-044 - Check the abort write flag after reads. STS 05-Aug-1982
 90 0090 1 | 1-045 - Make use of the bit masks for the options word. STS 17-Aug-1982
 91 0091 1 | 1-046 - Fix up the references to the EDTSM symbols. JBS 23-Aug-1982
 92 0092 1 | 1-047 - Put EDT\$\$TST_EOB in line. STS 22-Sep-1982
 93 0093 1 | 1-048 - Change to use new addline/subline macro. STS 1-Oct-1982
 94 0094 1 | 1-049 - Put code for edt\$\$rng_posfrst in line. STS 11-Oct-1982
 95 0095 1 | 1-050 - Modify to use new compare macro. STS 10-Oct-1982
 96 0096 1 | 1-051 - Avoid str\$copy where possible to increase efficiency. STS 10-Nov-1982
 97 0097 1 | 1-052 - Add conditional for WPS support. JBS 10-Feb-1983
 98 0098 1 | 1-053 - Only set screen width if it has changed from original. SMB 17-Feb-1983
 99 0099 1 | 1-054 - Give a message if requested to do so. JBS 24-Mar-1983
 100 0100 1 | 1-055 - Set a flag if control C actually aborts something. JBS 08-Apr-1983
 101 0101 1 | 1-056 - Delete the output file if we fail to write a record. JBS 02-May-1983
 102 0102 1 | 1-057 - Rearrange the delete output file logic to make this module smaller
 on PDP-11s, and to always give the "no output file created" message
 when we delete the output file. JBS 02-May-1983
 103 0103 1 | 1-058 - Because of search lists on VMS, don't use the input file name. JBS 29-Jul-1983
 104 0104 1 | --
 105 0105 1 |
 106 0106 1 |
 107 0107 1 |
 108 0108 1 |
 109 0109 1 |

```
111      0110 1 XSBTTL 'Declarations'  
112      0111 1 !  
113      0112 1 ! TABLE OF CONTENTS:  
114      0113 1 !  
115      0114 1 !  
116      0115 1 REQUIRE 'EDTSRC:TRARDUNAM';  
117      0554 1 !  
118      0555 1 FORWARD ROUTINE  
119          WRITE FILE,  
120          EDTSSEXI_CMD : NOVALUE,  
121          EDTSSWR_CMD : NOVALUE;  
122      0556 1 ! Write an output file  
123      0557 1 ! Process the EXIT command  
124      0558 1 ! Process the WRITE command  
125      0559 1 !  
126      0560 1 !  
127      0561 1 ! INCLUDE FILES:  
128      0562 1 !  
129      0563 1 !  
130      0564 1 REQUIRE 'EDTSRC:EDTREQ';  
131      0699 1 !  
132      L 0700 1 XIF %BLISS (BLISS32)  
133      0701 1 XTHEN  
134      0702 1 !  
135      0703 1 REQUIRE 'EDTSRC:SYSSYM';  
136      0733 1 !  
137      0734 1 XFI  
138      0735 1 !  
139      0736 1 LIBRARY 'EDTSRC:SUPPORTS';  
140      0737 1 !  
141      0738 1 ! MACROS:  
142      0739 1 !  
143      0740 1 !  
144      0741 1 !  
145      0742 1 !  
146      0743 1 ! EQUATED SYMBOLS:  
147      0744 1 !  
148      0745 1 !  
149      0746 1 EXTERNAL LITERAL  
150      0747 1 EDTSM_NOOUTPUT,  
151      0748 1 EDTSK_WRITE_FILE,  
152      0749 1 EDTSK_OUTPUT_FILE,  
153      0750 1 EDTSK_OPEN_OUTPUT_SEQ,  
154      0751 1 EDTSK_OPEN_OUTPUT_NOSEQ,  
155      0752 1 EDTSK_PUT,  
156      0753 1 EDTSK_CLOSE,  
157      0754 1 EDTSK_CLOSE_DEL;  
158      0755 1 !  
159      0756 1 ! OWN STORAGE:  
160      0757 1 !  
161      0758 1 !  
162      0759 1 !  
163      0760 1 !  
164      0761 1 ! EXTERNAL REFERENCES:  
165      0762 1 !  
166      0763 1 ! In the routines
```

165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221

0764 1 %SBTTL 'WRITE_FILE - write on a file'
0765 1 ROUTINE WRITE_FILE (
0766 1 RANGE,
0767 1 CHECK
0768 1) =
0769 1
0770 1 ++
0771 1 FUNCTIONAL DESCRIPTION:
0772 1
0773 1 This routine is used by the WRITE and EXIT commands to write
0774 1 an output file. The /SEQ switch determines whether sequence
0775 1 numbers should be written.
0776 1
0777 1 FORMAL PARAMETERS:
0778 1
0779 1 RANGE the range block for the range to be written.
0780 1
0781 1 CHECK a flag indicating that the consistency check should be done
0782 1 (set if we are exiting, clear otherwise). Also permits defaulting
0783 1 of the file name to the output or input file name.
0784 1
0785 1 IMPLICIT INPUTS:
0786 1
0787 1 EDTSSG_OUT_NAMLEN
0788 1 EDTSSA_OUT_NAM
0789 1 EDTSSG_INP_NAMLEN
0790 1 EDTSSA_INP_NAM
0791 1 EDTSSV_OPTIONS
0792 1 EDTSSA_CUR_BUF
0793 1 EDTSSL_ID_VFCHD
0794 1 EDTSSL_LNO_ZERO
0795 1 EDTSSL_LNO0
0796 1 EDTSSA_FMT_CUR
0797 1 EDTSST_FMT_BUF
0798 1 EDTSSG_FMT_LNPOS
0799 1 EDTSSL_LNO_VFCMAX
0800 1 EDTSSA_WK_CN
0801 1 EDTSSA_SUMRY
0802 1 EDTSSA_EXE_CURCMD
0803 1 EDTSSG_EXE_SBITS
0804 1 EDTSSZ_EXE_SBLK
0805 1 EDTSSG_WRITE_MSG
0806 1 EDTSSG_ABT_WRITE
0807 1
0808 1 IMPLICIT OUTPUTS:
0809 1
0810 1 EDTSSG_EXE_SBITS
0811 1 EDTSSG_CC_DONE
0812 1 EDTSSG_WRITE_MSG
0813 1 EDTSSG_ABT_WRITE
0814 1
0815 1 ROUTINE VALUE:
0816 1
0817 1 0 = no write took place
0818 1 1 = file written successfully
0819 1
0820 1 SIDE EFFECTS:

EDTSLWR
V04-000

EDTSLWRITE - write to a file
WRITE_FILE - write on a file

F 1
16-Sep-1984 01:03:36 VAX-11 Bliss-32 v4.0-742
14-Sep-1984 12:23:47 DISK\$VMSMASTER:[EDIT.SRC]LW

Page 5
(3)

ED
VO

```

222 0821 1 |
223 0822 1 | Types an error message if the file is not properly written.
224 0823 1 | If the consistency check fails, arranges to save the journal file.
225 0824 1 |
226 0825 1 | --
227 0826 1 |
228 0827 2 | BEGIN
229 0828 2 |
230 0829 2 | EXTERNAL ROUTINE
231 0830 2 |   EDTSSSC_SETWID,
232 0831 2 |   EDTSSSC_POSCSIF,
233 0832 2 |   EDTSSSTUP_WKINGMSG,
234 0833 2 |   EDTSSCHK_CC,
235 0834 2 |   EDTSSL_DIV,
236 0835 2 |   EDTSSCALLFIO,
237 0836 2 |   EDTSSFMT_CH,
238 0837 2 |   EDTSSFMT_STRCNT : NOVALUE,
239 0838 2 |   EDTSSFMT_CRLF,
240 0839 2 |   EDTSSFMT_MSG,
241 0840 2 |   EDTSSNXT_LNRNG,
242 0841 2 |   EDTSSRNG_REPOS,
243 0842 2 |   EDTSSFIOPN_ERR;
244 0843 2 |
245 L 0844 2 | %IF XBLISS (BLISS32)
246 0845 2 | %THEN
247 0846 2 |
248 0847 2 | EXTERNAL ROUTINE
249 0848 2 |   EDTSSFMT_STR,
250 0849 2 |   STRSFREET_DX;
251 0850 2 |
252 U 0851 2 | %ELSE
253 U 0852 2 |
254 U 0853 2 | EXTERNAL ROUTINE
255 U 0854 2 |   EDTSSGET_FNAME;
256 U 0855 2 |
257 U 0856 2 | EXTERNAL
258 U 0857 2 |   EDTSSG_INP_NAMLEN,
259 U 0858 2 |   EDTSSA_INP_NAM;
260 U 0859 2 |
261 ZFI 0860 2 |
262 0861 2 |
263 0862 2 | EXTERNAL
264 0863 2 |   EDTSSG_TI_WID,
265 0864 2 |   EDTSSG_ABORT_WRITE,
266 0865 2 |   EDTSSG_SAV_TIWID,
267 0866 2 |   EDTSSG_MESSAGE_LINE,
268 0867 2 |   EDTSSA_IO_FNAME,
269 0868 2 |   EDTSSG_EXT_MOD,
270 0869 2 |   EDTSSG_OUT_NAMLEN,
271 0870 2 |   EDTSSA_OUT_NAM,
272 0871 2 |   EDTSSG_RNG_FRSFLN,
273 0872 2 |   EDTSSZ_RNG_ORIGPOS : POS_BLOCK,
274 0873 2 |   EDTSSV_OPTIONS : BITVECTOR [32],
275 0874 2 |   EDTSSA_CUR_BUF : REF TBCB_BLOCK,
276 0875 2 |   EDTSSL_IO_VFCHD : WORD,
277 0876 2 |   EDTSSL_LNO_ZERO,
278 0877 2 |   EDTSSL_LNO_VFCMAX : LN_BLOCK,

```

! Reset terminal width
! Absolute cursor positioning
! Stop working message
! check for CONTROL/L typed
! sets up handler and calls general file routine
! Put a character in the format buffer
! Terminate an output line
! Put a message in the format buffer
! Print a file I/O error

! deallocates dynamic descriptors

! Length of input file name
! Address of input file name

! Terminal's width
! 1 = abort output

! Length of output file name
! Address of output file name

! bit 3 = /NOOUTPUT

! Value 6.5535 * (10**9)

```

279      0878 2      EDTSSL_LN00 : LNOVECTOR [14].           ! powers of ten
280      0879 2      EDTSSA_FMT_CUR.                   ! current position in format buffer
281      0880 2      EDTSSG_FMT_LNP0S.                 ! address of format buffer
282      0881 2
283      0882 2
284      L 0883 2      %IF SUPPORT_WPS
285      0884 2      %THEN
286      0885 2      EDTSSG_SUMRY.                  ! flag indicating whether to type out summary
287      0886 2      %FI
288      0887 2
289      0888 2      EDTSSZ_EOB_LN.                 ! pointer to the current command.
290      0889 2      EDTSSA_WK [N : REF LIN_BLOCK,    The options switches.
291      0890 2      EDTSSA_EXE_CURCMD : REF NODE_BLOCK, The option switch value block.
292      0891 2      EDTSSG_EXE_SBITS.                Set to 1 if control C actually aborts something
293      0892 2      EDTSSZ_EXE_SBLK : REF NODE_BLOCK,
294      0893 2      EDTSSG_CC_DONE.                 Message to print; 1 = no message
295      0894 2      EDTSSG_WRITE_MSG;
296      0895 2
297      P 0896 2      MESSAGES ((NOFILSPC, NOFILWRT, CONCHKFLD, ERROUTFIL, WRIFILCRE, OUTFILCRE, SEQNUMOV, SEQINCOV, OUTFILCL
298      0897 2      WRIFILCLO));
299      0898 2
300      0899 2      LOCAL
301      0900 2      CLOSEMSG.                    File close message
302      0901 2      OPNMSG.                      File open message
303      0902 2      FILECODE.                     code for what kind of open
304      0903 2      FILESTRM.                    code for which kind of file
305      0904 2      FILE_DESC : BLOCK [8, BYTE]. descriptor for file name
306      0905 2      RHB_DESC : BLOCK [8, BYTE]. descriptor for header info
307      0906 2      STAT_DESC : BLOCK [8, BYTE]. descriptor for puts
308      0907 2      STATUS.                      Status of open or close attempt
309      0908 2      L_COUNT : LN_BLOCK.       Number of lines written.
310      0909 2      C_COUNT.                     Number of characters written
311      0910 2      NAME.                        File name string pointer
312      0911 2      NAME_LEN.                    File name string length
313      0912 2      DIGIT.                       Holds integer result from division
314      0913 2      LINNO : LN_BLOCK.       48-bit line number
315      0914 2      SEQ.                         Flag indicating sequenced
316      0915 2      SEQ_NUM.                     Current sequence number
317      0916 2      SEQ_INCR:                   Sequence number increment
318      0917 2
319      L 0918 2      %IF %BLISS (BLISS32)
320      0919 2      %THEN
321      0920 2      RHB_DESC [DSCSB_DTYPE] = DSC$K_DTYPE_T;
322      0921 2      RHB_DESC [DSCSB_CLASS] = DSC$K_CLASS_S;
323      0922 2      STAT_DESC [DSCSB_DTYPE] = DSC$R_DTYPE_T;
324      0923 2      STAT_DESC [DSCSB_CLASS] = DSC$K_CLASS_S;
325      0924 2      FILE_DESC [DSCSB_DTYPE] = DSC$K_DTYPE_T;
326      0925 2      FILE_DESC [DSCSB_CLASS] = DSC$K_CLASS_D;
327      0926 2      %FI
328      0927 2
329      0928 2      RHB_DESC [DSCSA_POINTER] = 0;
330      0929 2      RHB_DESC [DSCSW_LENGTH] = 0;
331      0930 2      FILE_DESC [DSCSA_POINTER] = 0;
332      0931 2      FILE_DESC [DSCSW_LENGTH] = 0;
333      0932 2      !
334      0933 2      NAME_LEN = 0;
335      0934 2      !

```

```
336      0935 2 ! Position to the front of the range.  
337      0936 2 !-  
338      0937 2 EDTSSG RNG FRSTLN = 1;  
339      0938 2 EDTSSCPY_MEM (POS_SIZE, .EDTSSA_CUR_BUF, EDT$$Z_RNG_ORIGPOS);  
340      0939 2  
341      0940 2 IF ( NOT EDT$$RNG_REPOS (.RANGE)) THEN RETURN (0);  
342      0941 2  
343      0942 2 !+ Check for an explicit file specification.  
344      0943 2 !-  
345      0944 2  
346      0945 2  
347      0946 2 IF (.EDTSSA_EXE_CURCMD [FSPCLEN] NEQ 0)  
348      0947 2 THEN  
349      0948 2 BEGIN  
350      0949 2     NAME = .EDTSSA_EXE_CURCMD [FILSPEC];  
351      0950 2     NAME_LEN = .EDTSSA_EXE_CURCMD [FSPCLEN];  
352      0951 2 END;  
353      0952 2  
354      0953 2 !+  
355      0954 2 If no file name is specified and this is EXIT, use the output file name.  
356      0955 2 !-  
357      0956 2  
358      0957 2 IF ((.CHECK NEQ 0) AND (.NAME_LEN EQL 0))  
359      0958 2 THEN  
360      0959 2 BEGIN  
361      0960 2     NAME = .EDTSSA_OUT_NAM;  
362      0961 2     NAME_LEN = .EDTSSG_OUT_NAMLEN;  
363      0962 2 END;  
364      0963 2  
365      0964 2 !+  
366      0965 2 On VMS, because of search lists, it is possible for an input open to resolve  
367      0966 2 to a different directory than an output open with the same file name. Therefore,  
368      0967 2 if we want the output file to be returned to the same directory as the input file  
369      0968 2 we must depend on using the resultant file name from the input open to specify  
370      0969 2 where to put the output file, rather than being able to just re-use the input  
371      0970 2 file name.  
372      0971 2 At this point we reject the file specification only if it is empty and either  
373      0972 2 this is a WRITE command or /NOOUTPUT has been specified (or implied by /READ_ONLY).  
374      0973 2 !-  
375      0974 2  
376      0975 2 IF ((.NAME_LEN EQL 0) AND ((.CHECK EQL 0) OR ((.EDTSSV_OPTIONS AND EDT$M_NOOUTPUT) NEQ 0)))  
377      0976 2 THEN  
378      0977 2 BEGIN  
379      0978 2     EDT$SFMT MSG (EDTS_NOFILSPC);  
380      0979 2     RETURN (0);  
381      0980 2 END;  
382      0981 2  
383      0982 2 !+  
384      0983 2 PDP-11 systems do not have search lists, so we can use the input file name if the  
385      0984 2 output name is empty. Doing this simplifies the process of constructing the printable  
386      0985 2 form of the name for the summary message.  
387      0986 2 !-  
388      0987 2  
389      L 0988 2 %IF ( NOT %BLISS (BLISS32))  
390      U 0989 2 %THEN  
391      U 0990 2  
392      U 0991 2     IF (.NAME_LEN EQL 0)
```

```
393      U 0992 2      THEN
394      U 0993 2      BEGIN
395      U 0994 2      NAME = .EDTSSA_INP_NAM;
396      U 0995 2      NAME_LEN = .EDTSSG_INP_NAMLEN;
397      U 0996 2      END;
398      U 0997 2
399      0998 2      XFI
400      1000 2      :+ Check for the /SEQ option.
401      1001 2      :-+
402      1002 2      IF (SEQ = .EDTSSG_EXE_SBITS<OPB_SEQ>)
403      1003 2      THEN
404      1004 2      BEGIN
405      1005 2      :+ Determine the actual sequence start and increment by reducing them by 10**5
406      1006 2      :-
407      1007 2      SEQ_NUM = 0;
408      1008 2      SEQ_INCR = 0;
409      1009 2      MOVELINE (EDTSSZ_EXE_SBLK [SW_VAL1], LINNO); ! Starting Value
410      1010 2      :-+
411      1011 2      Error checks for sequence number and increment (must be less than 65536)
412      1012 2      :-+
413      1013 2      IF (CMPLNO (LINNO, EDTSSL_LNO_VFCMAX) GTR 0)
414      1014 2      THEN
415      1015 2      BEGIN
416      1016 2      EDTSSFMT_MSG (EDTS_SEQNUMOV);
417      1017 2      RETURN (0)
418      1018 2      END;
419      1019 3      DECR I FROM 9 TO 5 DO
420      1020 3      BEGIN
421      1021 3      EDTSSL DIV (LINNO, DIGIT, I);
422      1022 3      SEQ_NUM = .SEQ_NUM + .DIGIT;
423      1023 3      END;
424      1024 3      MOVELINE (EDTSSZ_EXE_SBLK [SW_VAL2], LINNO); ! Increment
425      1025 3      IF (CMPLNO (LINNO, EDTSSL_LNO_VFCMAX) GTR 0)
426      1026 3      THEN
427      1027 3      BEGIN
428      1028 3      EDTSSFMT_MSG (EDTS_SEQINCOV);
429      1029 3      RETURN (0)
430      1030 3      END;
431      1031 3      DECR I FROM 9 TO 5 DO
432      1032 3      BEGIN
433      1033 3      SEQ_INCR = .SEQ_INCR*10;
434      1034 3      EDTSSL DIV (LINNO, DIGIT, I);
435      1035 3      SEQ_INCR = .SEQ_INCR + .DIGIT;
436      1036 3      END;
437      1037 3      MOVELINE (EDTSSZ_EXE_SBLK [SW_VAL3], LINNO); ! Increment
438      1038 3      IF (CMPLNO (LINNO, EDTSSL_LNO_VFCMAX) GTR 0)
439      1039 3      THEN
440      1040 3      BEGIN
441      1041 3      EDTSSFMT_MSG (EDTS_SEQINCOV);
442      1042 3      RETURN (0)
443      1043 3      END;
444      1044 3      DECR I FROM 9 TO 5 DO
445      1045 3      BEGIN
446      1046 3      SEQ_INCR = .SEQ_INCR*10;
447      1047 3      EDTSSL DIV (LINNO, DIGIT, I);
448      1048 3      SEQ_INCR = .SEQ_INCR + .DIGIT;
449      1049 2      END;
```

```
450 1049 2 /*+  
451 1050 2 : Try to open the file and set up file specific messages  
452 1051 2 :-  
453 1052 2  
454 1053 2 IF (.CHECK EQ 0)  
455 1054 2 THEN  
456 1055 2 BEGIN  
457 1056 2 FILESTRM = EDTSK_WRITE_FILE;  
458 1057 2 CLOSEMSG = EDTS_QRIFILCLO;  
459 1058 2 OPNMSG = EDTS_Wrifilcre;  
460 1059 2 END  
461 1060 2 ELSE  
462 1061 2 BEGIN  
463 1062 2 FILESTRM = EDTSK_OUTPUT_FILE;  
464 1063 2 CLOSEMSG = EDTS_Outfilco;  
465 1064 2 OPNMSG = EDTS_Outfilcre;  
466 1065 2 END;  
467 1066 2  
468 1067 2 IF (.SEQ NEQ 0) THEN FILECODE = EDTSK_OPEN_OUTPUT_SEQ ELSE FILECODE = EDTSK_OPEN_OUTPUT_NOSEQ;  
469 1068 2  
470 1069 2 STRING_DESC (FILE_DESC, NAME_LEN, :NAME);  
471 1070 2 STATUS = EDTSSCAL[FIO (.FILECODE, :FILESTRM, FILE_DESC, RHB_DESC);  
472 1071 2  
473 1072 2 IF .STATUS  
474 1073 2 THEN  
475 1074 2 BEGIN  
476 1075 2 MOVELINE (EDTSSL_LNO_ZERO, L_COUNT);  
477 1076 2 C_COUNT = 0;  
478 1077 2 EDTSSG_ABT_WRITE = 0;  
479 1078 2  
480 1079 3 WHILE (EDTSSNXT_LNRNG (0) AND (.EDTSSA_WK_LN NEQA EDTSSZ_EOB_LN) AND ( NOT .EDTSSG_ABT_WRITE)) DO  
481 1080 4 BEGIN  
482 1081 4 /*+  
483 1082 4 : Check for a CONTROL/C. If one has been typed abort the write operation.  
484 1083 4 :-  
485 1084 4  
486 1085 4 IF EDTSSCHK_CC ()  
487 1086 4 THEN  
488 1087 5 BEGIN  
489 1088 5 EDTSSG_CC_DONE = 1;  
490 1089 5 EDTSSG_ABT_WRITE = 1;  
491 1090 5 END  
492 1091 4 ELSE  
493 1092 4 BEGIN  
494 1093 5 /*+  
495 1094 5 : Set up the RHB if /SEQ was used.  
496 1095 5 :-  
497 1096 5  
498 1097 6 IF (.SEQ NEQ 0)  
499 1098 6 THEN  
500 1099 5  
501 1100 6 IF (.EDTSSZ_EXE_SBLK [SEQ_VAL] EQ 0) !  
502 1101 5 THEN  
503 1102 5  
504 1103 5 /*+  
505 1104 5 : If no sequence start was given, then use the buffer's current line  
506 1105 5 numbers in the VFC field (/SEQ )
```

```
507      1106   6
508      1107   6
509      1108   6
510      1109   6
511      1110   6
512      1111   6
513      1112   6
514      1113   6
515      1114   7
516      1115   6
517      1116   6
518      1117   6
519      1118   7
520      1119   7
521      1120   7
522      1121   7
523      1122   7
524      1123   7
525      1124   7
526      1125   8
527      1126   8
528      1127   8
529      1128   8
530      1129   8
531      1130   8
532      1131   7
533      1132   7
534      1133   6
535      1134   5
536      1135   5
537      1136   5
538      1137   5
539      1138   6
540      1139   6
541      1140   6
542      1141   6
543      1142   6
544      1143   6
545      1144   5
546      1145   5
547      1146   5
548      1147   5
549      1148   5
550      1149   5
551      1150   5
552      1151   5
553      1152   5
554      1153   5
555      1154   5
556      1155   6
557      1156   5
558      1157   6
559      1158   6
560      L 1159   6
561      1160   6
562      1161   6
563      1162   6

      BEGIN
      MOVELINE (EDTSSA_WK_LN [LIN_NUM], LINNO);
      DECR I FROM 14 TO 10 DO
        WHILE (CMPLNO (LINNO, EDTSSL_LNOO [.I]) GEQ 0) DO
          SUBLINE (EDTSSL_LNOO [.I], LINNO);
        IF (CMPLNO (LINNO, EDTSSL_LNO_VFMAX) GEQ 0)
        THEN
          EDTSSL_IO_VFCHD = 65535
        ELSE
          BEGIN
            !+ Set up sequence numbers, but first divide by 10**5
            EDTSSL_IO_VFCHD = 0;
            DECR I FROM 9 TO 5 DO
              BEGIN
                EDTSSL_IO_VFCHD = .EDTSSL_IO_VFCHD*10;
                EDTSSL_DIV(LINNO, DIGIT, .I);
                EDTSSL_IO_VFCHD = .EDTSSL_IO_VFCHD + .DIGIT;
              END
            END
          ELSE
            !+ Otherwise, use the given sequence start and increment (/SEQ:st:inc)
            BEGIN
              EDTSSL_IO_VFCHD = .SEQ_NUM;
              SEQ_NUM = .SEQ_NUM + .SEQ_INCR;
              IF (.SEQ_NUM EQL 65535) THEN SEQ_INCR = 0;
            END;
          !+ Write a line to the file.
          RHB_DESC [DSCSW_LENGTH] = 2;
          RHB_DESC [DSCSA_POINTER] = EDTSSL_IO_VFCHD;
          STAT_DESC [DSCSQ_LENGTH] = .EDTSSA_WR_LN [LIN_LENGTH];
          STAT_DESC [DSCSA_POINTER] = EDTSSA_WK_LN [LIN_TEXT];
          STATUS = EDTSSCA[LFIO (EDTSK_PUT, .FI[ESTRM, STAT_DESC, RHB_DESC]);
          IF ( NOT .STATUS)
          THEN
            BEGIN
              !+ XIF XBLISS (BLISS32)
              !+ THEN
              EDTSSFIOPN_ERR (EDTS_ERROUTFIL, .EDTSSA_IO_FNAM);
              !+ FI
            END
          END
        END
      END
    END
  END
END
```

```
564      1163 6
565      1164 6
566      1165 6      EDTSSG_ABT_WRITE = 1;
567      1166 6      END
568      1167 6      ELSE BEGIN
569      1168 6      |+ Count lines and characters for consistency check.
570      1169 6      |-
571      1170 6      ADDLINE (NUMBER ONE, L_COUNT);
572      1171 6      C_COUNT = .C_COUNT + .EDTSSA_WK_LN [LIN_LENGTH];
573      1172 6      END;
574      1173 5
575      1174 5
576      1175 4      END;
577      1176 4
578      1177 3      END;
579      1178 3
580      1179 3      |+
581      1180 3      |+ Watch for the write abort flag, which can be set if we have an error on the input file,
582      1181 3      |+ by control C or by an error writing the output file.
583      1182 3      |-
584      1183 3
585      1184 3      IF .EDTSSG_ABT_WRITE
586      1185 3      THEN
587      1186 4      BEGIN
588      1187 4      STATUS = EDTSSCALLFIO (EDTSK_CLOSE_DEL, .FILESTRM, FILE_DESC, 0);
589      1188 4
590      1189 4      IF ( NOT .STATUS) THEN EDTSSFIOPN_ERR (.CLOSEMSG, FILE_DESC);
591      1190 4
592      1191 4      EDTSSFMT_MSG (EDTS_NOFILWRT);
593      1192 4
594      L 1193 4      %IF %BLISS (BLISS32)
595      1194 4      %THEN
596      1195 4      STR$FREE1_DX (FILE_DESC);
597      1196 4      %FI
598      1197 4
599      1198 4      RETURN (0);
600      1199 3      END;
601      1200 3
602      1201 3      |+
603      1202 3      |+ Do the consistency check.
604      1203 3      |-
605      1204 3
606      1205 4      IF (.CHECK NEQ 0)
607      1206 4      THEN
608      1207 4
609      1208 4      IF ( NOT (LINNOEQL (L_COUNT, EDTSSA_CUR_BUF [TBCB_LINE_COUNT])) OR !
610      1209 4      (.C_COUNT NEQ .EDTSSA_CUR_BUF [TBCB_CHAR_COUNT]))
611      1210 3      THEN
612      1211 4      BEGIN
613      1212 4      EDTSSFMT_MSG (EDTS_CONCHKFLD);
614      1213 4      EDTSSG_EXE_SBITS = (.EDTSSG_EXE_SBITS OR OPT_SAVE);
615      1214 3      END;
616      1215 3
617      1216 3      |+
618      1217 3      |+ Close the output file and print a message giving number of lines written
619      1218 3      to the file.
620      1219 3      |-
```

```
1220      3      IF (.EDTSSG_EXT_MOD) THEN EDTSSSTOP_WKINGMSG ();  
1221      3      STATUS = EDTSSCALLFIO (EDT$K_CLOSE, .FILESTRM, FILE_DESC, 0);  
1222      3      IF .STATUS  
1223      3      THEN  
1224      3      BEGIN  
1225      3      !+ Reset the screen width on EXIT if necessary. If the screen was reset, then  
1226      3      reposition the cursor at the bottom of the screen.  
1227      3      !-  
1228      4      IF (.CHECK NEQ 0)  
1229      4      THEN  
1230      4      !+  
1231      4      IF (.EDTSSG_TI_WID NEQ .EDTSSG_SAV_TIWID)  
1232      4      THEN  
1233      4      BEGIN  
1234      4      EDTSSSC_SETWID (.EDTSSG_SAV_TIWID);  
1235      4      EDTSSSC_POSCSIF (.EDTSSG_MESSAGE_LINE, 0);  
1236      4      END;  
1237      4      !+  
1238      4      Print a message if one is requested. This will be the 'input file does not have standard format'  
1239      4      message deferred because we thought the user was never going to write the buffer.  
1240      4      !-  
1241      4      IF (.EDTSSG_WRITE_MSG NEQ 1)  
1242      4      THEN  
1243      4      BEGIN  
1244      4      EDTSSFMT_MSG (.EDTSSG_WRITE_MSG);  
1245      4      EDTSSG_WRITE_MSG = 1;  
1246      4      END;  
1247      4      !+  
1248      5      IF SUPPORT_WPS  
1249      5      THEN  
1250      5      !+  
1251      5      IF .EDTSSG_SUMRY  
1252      5      THEN  
1253      5      BEGIN  
1254      5      !+  
1255      5      Extract the resultant filename of the opened file and format  
1256      5      it to write out if the summary flag is set.  
1257      5      !-  
1258      5      !+  
1259      5      IF .EDTSSG_SUMRY  
1260      5      THEN  
1261      5      !+  
1262      5      BEGIN  
1263      5      !+  
1264      5      Extract the resultant filename of the opened file and format  
1265      5      it to write out if the summary flag is set.  
1266      5      !-  
1267      5      !+  
1268      5      IF #BLISS (BLISS32)  
1269      5      THEN  
1270      5      EDTSSFMT_STR (.FILE_DESC [DSCBA_POINTER], .FILE_DESC [DSCSW_LENGTH]);  
1271      5      ELSE  
1272      5      EDTSSGET_FNAME ();  
1273      5      !+  
1274      5      EDTSSFMT_CH (' ');  
1275      5      EDTSSFMT_STRCNT (L_COUNT, UPLIT ('Line'), 5);  
1276      5
```

```
678 1277 5          EDTSSFMT_CRLF ()  
679 1278 4          END;  
680 1279 4  
681 1280 4 !+  
682 1281 4 ! On VMS deallocate all dynamic descriptors used  
683 1282 4 !-  
684 1283 4  
685 L 1284 4 %IF %BLISS (BLISS32)  
686 1285 4 %THEN  
687 1286 4     STR$FREE1_DX (FILE_DESC);  
688 1287 4 %FI  
689 1288 4  
690 1289 4     RETURN (1);  
691 1290 4     END  
692 1291 3 ELSE  
693 1292 4 BEGIN  
694 1293 4 !+  
695 1294 4 ! File was not closed, output error message.  
696 1295 4 !-  
697 1296 4     EDT$SFOPEN_ERR (.CLOSEMSG, FILE_DESC);  
698 1297 4 !+  
699 1298 4 ! On VMS deallocate all dynamic descriptors used  
700 1299 4 !-  
701 1300 4  
702 L 1301 4 %IF %BLISS (BLISS32)  
703 1302 4 %THEN  
704 1303 4     STR$FREE1_DX (FILE_DESC);  
705 1304 4 %FI  
706 1305 4  
707 1306 4     RETURN (0);  
708 1307 4     END  
709 1308 4  
710 1309 3 END  
711 1310 2 ELSE  
712 1311 2 BEGIN  
713 1312 4 !+  
714 1313 4 ! File was not created, output error message.  
715 1314 4 !-  
716 1315 4     EDT$SFOPEN_ERR (.OPNMSG, FILE_DESC);  
717 1316 4 !+  
718 1317 4 ! On VMS deallocate all dynamic descriptors used  
719 1318 4 !-  
720 1319 4  
721 L 1320 4 %IF %BLISS (BLISS32)  
722 1321 4 %THEN  
723 1322 4     STR$FREE1_DX (FILE_DESC);  
724 1323 4 %FI  
725 1324 4  
726 1325 4     RETURN (0);  
727 1326 4     END  
728 1327 4  
729 1328 1 END;
```

! of routine WRITE_FILE

:TITLE EDTSLWRITE EDTSLWRITE - write to a file
.IDENT \V04-000\

EDTSLWRITE
V04-000

EDTSLWRITE - write to a file
WRITE_FILE - write on a file

B 2

16-Sep-1984 01:03:36
14-Sep-1984 12:23:47

VAX-11 BLiss-32 V4.0-742
DISK\$VMSMASTER:[EDT.SRC]LWRITE.BLI;1

Page 14
(3)

00 00 00 65 6E 69 6C 20 00000 P.AAA:

.PSECT _EDT\$CODE,NOWRT, SHR, PIC,2

.ASCII \ Line\<0><0><0>

.EXTRN EDTSM_NOOUTPUT, EDTSK_WRITE_FILE
.EXTRN EDTSK_OUTPUT_FILE
.EXTRN EDTSK_OPEN_OUTPUT_SEQ
.EXTRN EDTSK_OPEN_OUTPUT_NOSEQ
.EXTRN EDTSK_PUT, EDTSK_CLOSE
.EXTRN EDTSK_CLOSE_DEL
.EXTRN EDTSSSC_SETVID, EDTSSSC_POSCSIF
.EXTRN EDTSSSTOP_WKINGMSG
.EXTRN EDTSSCHK_CC, EDTSSL DIV
.EXTRN EDTSSCALEFI, EDTSSFMT_CH
.EXTRN EDTSSFMT_STRCNT
.EXTRN EDTSSFMT_CRLF, EDTSSFMT_MSG
.EXTRN EDTSSNXT_LNRNG, EDTSSRNG_REPOS
.EXTRN EDTSSF_IOPN_ERR, EDTSSFMT_STR
.EXTRN STR\$FREE1 BX, EDTSSG_TI_WID
.EXTRN EDTSSG_ABZ_WRITE
.EXTRN EDTSSG_SAV_TIWID
.EXTRN EDTSSG_MESSAGE_LINE
.EXTRN EDTSSA_IO_FNAME, EDTSSG_EXT_MOD
.EXTRN EDTSSG_OUT_NAMLEN
.EXTRN EDTSSA_OUT_NAM, EDTSSG_RNG_FRSTLN
.EXTRN EDTSSZ_RNG_ORIGPOS
.EXTRN EDTSSV_OPTIONS, EDTSSA_CUR_BUF
.EXTRN EDTSSL_IO_VFCHD
.EXTRN EDTSSL_LNO_ZERO
.EXTRN EDTSSL_LNO_VFCMAX
.EXTRN EDTSSL_LNO, EDTSSA_FMT_CUR
.EXTRN EDTSSG_FMT_LNPOS
.EXTRN EDTSST_FMT_BUF, EDTSSG_SUMRY
.EXTRN EDTSSZ_EOB_LN, EDTSSA_WK_LN
.EXTRN EDTSSA_EXE_CURCMD
.EXTRN EDTSSG_EXE_SBITS
.EXTRN EDTSSZ_EXE_SBLK
.EXTRN EDTSSG_CC_DONE, EDTSSG_WRITE_MSG
.EXTRN EDTS_NOFI_SPC, EDTS_NOFILWRT
.EXTRN EDTS_CONCHKFLD, EDTS_ERROUTFIL
.EXTRN EDTS_WRIFILCRE, EDTS_OUTFILCRE
.EXTRN EDTS_SEQNUMOV, EDTS_SEQINCROV
.EXTRN EDTS_OUTFILCLO, EDTS_WRIFILCLO
.EXTRN STR\$COPY_R

OFFC 00000 WRITE_FILE:

28	AE	010E0000	38	C2	00002	WORD	Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11	: 0765
22	AE	010E	8F	D0	00005	SUBL2	#56, SP	: 0929
30	AE	020E0000	8F	D0	0000D	MOVW	#17694720, RHB_DESC	: 0922
		2C	AE	D4	0001B	MOVL	#270, STAF_DESC+2	: 0931
		34	AE	D4	0001E	CLRL	#34471936, FILE_DESC	: 0928
		08	AE	D4	00021	CLRL	RHB_DESC+4	: 0930
		00000000G	00	01	D0	00024	FILE_DESC+4	: 0933
		00000000G	50	00000000G	00	0002B	NAME_LEN	: 0937
00000000G	00		60	0E	28	00032	#1, EDTSSG_RNG_FRSTLN	: 0938
						MOVLC3	EDTSSA_CUR_BUF_R0	:
							#14, (R0), EDTSSZ_RNG_ORIGPOS	:

**EDTSLWRITE
V04-000**

EDTSLWRITE - write to a file
WRITE_FILE - write on a file

C 2
16-Sep-1984 01:03:36 VAX-11 BLISS-32 V4.0-742
14-Sep-1984 12:23:47 DISK\$VMSMASTER:[EDIT.SRC]LWRITE.BLI:1

Page 15
(3)

ED
VO

							RANGE		0940	
00000000G	00	04	AC	DD	0003A	PUSHL	#1, EDTSSRNG_REPO\$			
	03		01	FB	0003D	CALLS	R0, 1\$			
		50	F8	00044		BLBS	68\$			
	50	0484	31	00047	18:	BRW	12(R0)			
	50 00000000G	00	D0	0004A	18:	MOVL	EDTSSA_EXE_CURCMD, R0		0946	
	0C	A0	D5	00051		TSTL				
		09	13	00054		BEQL	2\$			
08	5A	08	A0	D0	00056	MOVL	8(R0), NAME		0949	
	AE	0C	A0	D0	0005A	MOVL	12(R0), NAME_LEN		0950	
	56	08	AC	D0	0005F	MOVL	CHECK, R6		0957	
		04	AE	D4	00063	CLRL	4(SP)			
			56	D5	00066	TSTL	R6			
			17	13	00068	BEQL	3\$			
		04	AE	D6	0006A	INCL	4(SP)			
		08	AE	D5	0006D	TSTL	NAME_LEN			
			0F	12	00070	BNEQ	3\$			
08	5A 00000000G	00	D0	00072		MOVL	EDTSSA_OUT_NAM, NAME		0960	
	AE 00000000G	00	D0	00079		MOVL	EDTSSG_OUT_NAMLEN, NAME_LEN		0961	
		08	AE	D5	00081	TSTL	NAME_LEN		0975	
			19	12	00084	BNEQ	5\$			
			56	D5	00086	TSTL	R6			
			0D	13	00088	BEQL	4\$			
00000000G	8F 00000000G	00	D3	0008A		BITL	EDTSSV_OPTIONS, #EDTSM_NOOUTPUT			
		08	13	00095		BEQL	5\$			
		00000000G	8F	DD	00097	48:	PUSHL	#EDTS_NOFILESPC		
			52	11	0009D	BRB	11\$		0978	
57 00000000G	00	01	04	EF	0009F	EXTZV	#4, #1, EDTSSG_EXE_SBITS, SEQ			
		03	57	E8	000A8	BLBS	SEQ, 6\$		1004	
			00CE	31	000AB	BRW	21\$			
			59	D4	000AE	CLRL	SEQ_NUM		1010	
			5B	D4	000B0	CLRL	SEQ_INCR		1011	
10	AE	08	50 00000000G	00	D0	000B2	MOVL	EDTSSZ_EXE_SBLK, R0		1012
			A0	06	28	000B9	MOVC3	#6, 8(R0), LINNO		
			50 00000000G	00	3C	000BF	MOVZWL	HIGH_2, R0		
			50	14	AE	B1	CMPW	HIGH_1, R0		1017
				0F	1F	000CA	BLSSU	7\$		
				18	12	000CC	BNEQ	9\$		
			50 00000000G	00	D0	000CE	MOVL	LOW_2, R0		
			50	10	AE	D1	Cmpl	LOW_1, R0		
				05	1E	000D9	BGEQU	8\$		
			50	01	CE	000DB	MNEGGL	#1, R0		
				09	11	000DE	BRB	10\$		
				04	12	000E0	BNEQ	9\$		
				50	D4	000E2	CLRL	R0		
				03	11	000E4	BRB	10\$		
			50	01	D0	000E6	MOVL	#1, R0		
				08	15	000E9	BLEQ	12\$		
			00000000G	8F	DD	000EB	PUSHL	#EDTS_SEQNUMOV		1020
				5F	11	000F1	BRB	18\$		
			52	09	D0	000F3	MOVL	#9, I		1024
			59	0A	C4	000F6	MULL2	#10, SEQ_NUM		1026
				52	DD	000F9	PUSHL	I		1027
				10	AE	9F	PUSHAB	DIGIT		
				18	AE	9F	PUSHAB	LINNO		
00000000G	00	0C	03	F8	00101	CALLS	#3, EDTSSLDIV			1028
			59	AE	C0	00108	ADDL2	DIGIT, SEQ_NUM		
FFE3	52	FF	8F	05	9D	0010C	ACBB	#5, #1, I, 13\$		1024

EDTS WRITE
V04-000

EDTSLWRITE - write to a file
WRITE_FILE - write on a file

D 2
16-Sep-1984 01:03:36
14-Sep-1984 12:23:47

VAX-11 Bliss-32 v4.0-742
DISKSVMMASTER:[EDIT.SRC]LW

Page 16
(3)

E
8

10	AE	14	50 0000000G	00	D0 00113	MOVL	EDTSSZ_EXE_SBLK, RO		1031		
			A0 0000000G	06	28 0011A	MOV C3	#6_20TR0), LINNO				
			50 0000000G	00	3C 00120	MOVZWL	HIGH_2, RO		1033		
			50 14	AE	B1 00127	CMPW	HIGH_1, RO				
				0F	1F 0012B	BLSSU	14S				
			50 0000000G	18	12 0012D	BNEQ	16S				
			50 10	00	D0 0012F	MOVL	LOW_2, RO				
				AE	D1 00136	CMPYL	LOW_1, RO				
			50	05	1E 0013A	BGFQU	15S				
				01	CE 0013C	MNE SL	#1 RO				
				09	11 0013F	BRB	17S				
				04	12 00141	BNEQ	16S				
				50	D4 00143	CLKL	RO				
				03	11 00145	BRB	17S				
			50	01	D0 00147	MOVL	#1 RO				
				10	15 0014A	BLEQ	19S				
			0000000G	00	8F DD 0014C	PUSHL	NEDTS_SEQINCROV		1036		
				01	FB 00152	CALLS	#1 EDTSSFMT_MSG				
				0372	31 00159	BRW	68S		1037		
			52	09	D0 0015C	MOVL	#9 I		1040		
			5B	0A	C4 0015F	MULL2	#10, SEQ_INCR		1042		
				52	DD 00162	PUSHL	I		1043		
				10	AE 9F 00164	PUSHAB	DIGIT				
				18	AE 9F 00167	PUSHAB	LINNO				
			0000000G	00	03 FB 0016A	CALLS	#3, EDTSSLDIV		1044		
				5B	AE C0 00171	ADDL2	DIGIT, SEQ_INCR		1040		
				8F	05 9D 00175	ACBB	#5, #1, I, 20S		1053		
				0C	56 D5 0017C	TSTL	R6				
					17 12 0017E	BNEQ	22S				
			56 0000000G	8F	D0 00180	MOVL	NEDTSK_WRITE FILE, FILESTRM		1056		
			58 0000000G	8F	D0 00187	MOVL	NEDTS_WRIFILCLO, CLOSEMSG		1057		
			53 0000000G	8F	D0 0018E	MOVL	NEDTS_WRIFILCRE, OPNMSG		1058		
				15	11 00195	BRB	23S		1053		
			56 0000000G	8F	D0 00197	MOVL	NEDTSK_OUTPUT FILE, FILESTRM		1062		
			58 0000000G	8F	D0 0019E	MOVL	NEDTS_OUTFILCLO, CLOSEMSG		1063		
			53 0000000G	8F	D0 001A5	MOVL	NEDTS_OUTFILCRE, OPNMSG		1064		
				6E	D4 001AC	CLRL	(SP)		1067		
				57	D5 001AE	TSTL	SEQ				
				0B	13 001B0	BEQL	24S				
				08	13 001B0	INCL	(SP)				
			52 0000000G	8F	D0 001B4	MOVL	NEDTSK_OPEN_OUTPUT_SEQ, FILECODE				
				07	11 001BB	BRB	25S				
			52 0000000G	8F	D0 001BD	MOVL	NEDTSK_OPEN_OUTPUT_NOSEQ, FILECODE				
				0C	5A DD 001C4	24S:			1069		
				38	AE 9F 001C6	PUSHL	NAME				
				03	AE 9F 001C9	PUSHAB	NAME_LEN				
				28	FB 001CC	CALLS	FILE_DESC				
				34	AE 9F 001D3	PUSHAB	#3, STRSCOPY_R				
				0044	AE 9F 001D6	PUSHAB	RHB DESC		1070		
			0000000G	00	8F BB 001D9	PUSHAB	FILE DESC				
				04	FB 001DD	PUSHR	#^M<R2, R6>				
				57	D0 001E4	CALLS	#4, EDTSSCALLFIO				
				03	57 E8 001E7	MOVL	RO_STATUS				
				02CB	31 001EA	BLBS	STATUS, 26S				
				06	28 001ED	BRW	65S				
			0000000G	00	5A D4 001F6	26S:	MOVC3	#6, EDTSSL_LNO_ZERO, L_COUNT		1075	
				00	D4 001F8	CLRL	COUNT				
						CLRL	EDTSSG_AB_T_WRITE				

EDTSLWRITE
V04-000

EDT\$LWRITE = write to a file
WRITE_FILE = write on a file

E 2
16-Sep-1984 01:03:36 VAX-11 Bliss-32 v4.0-742
14-Sep-1984 12:23:47 DISK\$VMSMASTER:[EDT.SRC]LWRITE.BLI;

Page 17
(3)

ED
VC

Assembly Language Disassembly										Line Number
00000000G	00	7E	D4	001FE	27\$:	CLRL	-(SP)			1079
	03	01	FB	00200		CALLS	#1.	EDTSSNXT_LNRNG		
		50	F8	00207		BLBS	R0,	29\$		
50	00000000G	00	17D	31	0020A	28\$:	BRW	53\$		
50	00000000G	00	9E	00200	29\$:	MOVAB	EDT\$\$Z_EOB_LN, R0			
		D1	00214			CMPL	EDTSSA_WK_N, R0			
03	00000000G	00	ED	13	0021B		BEQL	28\$		
		F9	0021D			BLBC	EDTSSG_ABT_WRITE, 30\$			
00000000G	00	016A	31	00224		BRW	54\$			
		00	FB	00227	30\$:	CALLS	#0.	EDTSSCHK_CC		1085
	0A	50	E9	0022E		BLBC	R0,	31\$		
00000000G	00	01	D0	00231		MOVL	#1	EDTSSG_CC_DONE		1088
		012E	31	00238		BRW	49\$			1089
03		6E	F8	0023B	31\$:	BLBS	(SP),	32\$		1097
50	00000000G	00	0DE	31	0023E	32\$:	BRW	48\$		
		D0	00241			MOVL	EDT\$\$Z_EXE_SBLK, R0			1100
		A0	95	00248		TSTB	1(R0)			
		03	13	0024B		BEQL	33\$			
		00BA	31	0024D		BRW	47\$			
		00	D0	00250	33\$:	MOVL	EDTSSA_WK_LN, R0			1107
10	AE	50	00000000G	00	00257		MOV3	#6, 1(R0), LINNO		
		A0	06	00250		MOVL	#14, I			1109
51		50	0E	00250		MULL3	#6, I, R1			1111
		50	06	C5	00260	34\$:	MOVAB	EDTSSL_LN00[R1], R1		
		51	00000000G	0041	9E	00264	CMPW	HIGH_1, 4(R1)		
		A1	14	AE	B1	0026C	35\$:	BLSSU	36\$	
			08	1F	00271		BNEQ	38\$		
			11	12	00273		CMPL	LOW_1, (R1)		
		61	10	AE	D1	00275	BGEQU	37\$		
			05	1E	00279	36\$:	MNEGGL	#1, R2		
		52	01	CE	0027B	36\$:	BRB	39\$		
			09	11	0027E	37\$:	BNEQ	38\$		
			04	12	00280	37\$:	CLRL	R2		
		52	52	D4	00282		BRB	39\$		
			03	11	00284		MOVL	#1, R2		
		52	01	D0	00286	38\$:	BLSS	40\$		
			13	19	00289	39\$:	MOVW	UPPER_WORD, SAVE		1112
		10	AE	80	0028B		SUBL2	(R1), LINNO		
		14	AE	61	C2	0028F	SBWC	4(R1), LINNO		
		16	AE	04	A1	D9	MOVW	SAVE, UPPER_WORD		1111
		16	AE	52	B0	00298	BRB	35\$		
			CE	11	0029C		ACBB	#10, #-1, I, 34\$		
50	FF	8F	0A	90	0029E	40\$:	MOVZWL	HIGH_2, R0		
		50	00000000G	00	3C	002A5	CMPW	HIGH_1, R0		1114
		50	14	AE	B1	002AC	BLSSU	41\$		
			0F	1F	002B0		BNEQ	43\$		
		50	00000000G	00	18	002B2	MOVL	LOW_2, R0		
		50	10	AE	D0	002B4	CMPL	LOW_1, R0		
			05	1E	002BB		BGEQU	42\$		
		50	01	CE	002C1	41\$:	MNEGGL	#1, R0		
			09	11	002C4		BRB	44\$		
			04	12	002C6	42\$:	BNEQ	43\$		
		50	50	D4	002C8		CLRL	R0		
			03	11	002CA		BRB	44\$		
		50	01	D0	002CC	43\$:	MOVL	#1, R0		
			09	19	002CF	44\$:	BLSS	45\$		
		00000000G	00	01	AE	002D1	MNEGWL	#1, EDTSSL_ID_VFCHD		1116

EDTSLWRITE
V04-000

EDTSLWRITE - write to a file
WRITE_FILE - write on a file

F 2
16-Sep-1984 01:03:36
14-Sep-1984 12:23:47

VAX-11 Bliss-32 V4.0-742
DISKSVMMASTER:[EDT.SRC]LWRITE.BLI:

Page 18
(3)

ED
VO

EDTSLWRITE
V04-000

EDTSLWRITE - write to a file
WRITE_FILE - write on a file

G 2
16-Sep-1984 01:03:36 VAX-11 Bliss-32 V4.0-742 Page 19
14-Sep-1984 12:23:47 DISKS\$VMSMASTER:[EDIT.SRC]LWRITE.BLI:1 (3)

ED
VO

1C	A0	1C	0D	12	003D7	BNEQ	57\$		1209	
1E	A0		06	12	003DE	CMPW	HIGH_1, 28(R0)			
		5A	D1	003EO	BNEQ	57\$				
		15	13	003E4	CMPL	C COUNT, 30(R0)				
00000000G	00	00000000G	8F	DD	003E6	BEQL	58\$			
00000000G	00	40	01	FB	003EC	PUSHL	#EDTS CONCHKFLD		1212	
00000000G	07	00000000G	8F	88	003F3	CALLS	#1, EDTSSFMMSG		1213	
00000000G	00		00	E9	003FB	BISB2	#64, EDTSSG EXE_SBITS		1221	
			00	FB	00402	BLBC	EDTSSG EXT MOD, 59\$			
		34	7E	D4	00409	CALLS	#0, EDTSSSTOP_WKINGMSG			
			AE	9F	0040B	CLRL	-(\$P)		1223	
			56	DD	0040E	PUSHAB	FILE DESC			
00000000G	00	00000000G	8F	DD	00410	PUSHL	FILESTRM			
57			04	FB	00416	PUSHL	#EDTSTK CLOSE			
03			50	DO	0041D	CALLS	#4, EDTSSCALLFIO			
			57	E8	00420	MOVL	RO, STATUS			
		008B	31	00423	BLBS	STATUS, 60\$			1225	
28	04	AE	E9	00426	BRW	64\$			1233	
50	00000000G	00	DO	0042A	BLBC	4(SP), 61\$			1236	
50	00000000G		00	D1	00431	MOVL	EDTSSG_SAV TIWID, RO			
			18	13	00438	CMPL	EDTSSG_TI_WID, RO			
00000000G	00		50	DD	0043A	BEQL	61\$			
			01	FB	0043C	PUSHL	RO		1239	
			7E	D4	00443	CALLS	#1, EDTSSSC_SETWID			
00000000G	00	00000000G	00	DD	00445	CLRL	-(\$P)		1240	
50	00000000G	00	02	FB	0044B	PUSHL	EDTSSG_MESSAGE LINE			
01			DO	00452	CALLS	#2, EDTSSSC_P05CSIF			1248	
			50	D1	00459	MOVL	EDTSSG_WRITE_MSG, RO			
			10	13	0045C	CMPL	RO, #1			
00000000G	00		50	DD	0045E	BEQL	62\$			
			01	FB	00460	PUSHL	RO		1251	
00000000G	00		01	DO	00467	CALLS	#1, EDTSSFMT_MSG		1252	
2E	00000000G	00	E9	0046E	62\$:	MOVL	#1, EDTSSG_WRITE_MSG		1258	
7E	30	AE	3C	00475	BLBC	EDTSSG_SUMRY, 63\$			1270	
	38	AE	DD	00479	MOVZWL	FILE_DESC, -(\$P)				
00000000G	00		02	FB	0047C	PUSHL	FILE_DESC+4			
			20	DD	00483	CALLS	#2, EDTSSFMT_STR		1275	
00000000G	00		01	FB	00485	PUSHL	#32			
			05	DD	0048C	CALLS	#1, EDTSSFMT_CH		1276	
		FB66	CF	9F	0048E	PUSHL	#5			
		20	AE	9F	00492	PUSHAB	P.AAA			
00000000G	00		03	FB	00495	PUSHAB	L_COUNT			
00000000G	00		00	FB	0049C	CALLS	#3, EDTSSFMT_STRCNT		1277	
00000000G	00	30	AE	9F	004A3	63\$::	CALLS	#0, EDTSSFMT_CRLF		1286
			01	FB	004A6	PUSHAB	FILE DESC			
	50		01	DO	004AD	CALLS	#1, STR\$FREE1_DX			
			04	004B0		MOVL	#1, RO		1292	
			30	AE	004B1	64\$::	RET			
			58	DD	004B4	PUSHAB	FILE DESC		1296	
			05	11	004B6	PUSHL	CLOSEMSG			
			30	AE	004B8	65\$::	BRB	66\$		
			53	DD	004BB	PUSHAB	FILE DESC		1315	
00000000G	00		02	FB	004BD	PUSHL	OPNMSG			
			01	FB	004C4	66\$::	CALLS	#2, EDTSSFIOPN_ERR		1322
00000000G	00		01	FB	004C7	PUSHAB	FILE DESC			
			50	D4	004CE	68\$::	CALLS	#1, STR\$FREE1_DX		1328
						CLRL	RO			

EDT\$LWRITE
V04-000

EDT\$LWRITE - write to a file
WRITE_FILE - write on a file

H 2
16-Sep-1984 01:03:36
12-Sep-1984 12:23:49

VAX-11 Bliss-32 v4.0-742
DISK\$VMSMASTER:[EDT.SRC]LWRITE.BLI;1 Page 20
(3)

ED
VO

04 00400 RET

; Routine Size: 1233 bytes, Routine Base: _EDT\$CODE + 0008

```
731      1329 1 %SBTTL 'EDTSSEXI_CMD - EXIT line-mode command'  
732      1330 1  
733      1331 1 GLOBAL ROUTINE EDTSSEXI_CMD  
734      1332 1 : NOVALUE =  
735      1333 1  
736      1334 1 ++  
737      1335 1 FUNCTIONAL DESCRIPTION:  
738      1336 1  
739      1337 1 Command processing routine for exit. Switch to the main buffer,  
740      1338 1 write the output file and set the EDTSSG_EXITD flag  
741      1339 1 if it succeeded.  
742      1340 1  
743      1341 1 FORMAL PARAMETERS:  
744      1342 1  
745      1343 1  
746      1344 1  
747      1345 1  
748      1346 1  
749      1347 1  
750      1348 1  
751      1349 1  
752      1350 1  
753      1351 1  
754      1352 1  
755      1353 1  
756      1354 1  
757      1355 1  
758      1356 1  
759      1357 1  
760      1358 1  
761      1359 1  
762      1360 1  
763      1361 1 --  
764      1362 1  
765      1363 2 BEGIN  
766      1364 2  
767      1365 2 EXTERNAL ROUTINE  
768      1366 2 EDT$SFND_BUF;  
769      1367 2  
770      1368 2  
771      1369 2 EXTERNAL  
772      1370 2 EDTSSG_EXITD,  
773      1371 2 EDTSSG_RCOV_MOD;  
774      1372 2 LOCAL  
775      1373 2 RANGE : NODE_BLOCK;  
776      1374 2  
777      1375 2 +  
778      1376 2 Ignore the EXIT command encountered during recovery.  
779      1377 2 -  
780      1378 2  
781      1379 2 IF .EDTSSG_RCOV_MOD THEN RETURN;  
782      1380 2  
783      1381 2 +  
784      1382 2 Setup the range as WHOLE.  
785      1383 2 -  
786      1384 2 RANGE [NODE_TYPE] = RANGE_NODE;  
787      1385 2 RANGE [RAN_TYPE] = RAN_WHOLE;
```

EDTSLWRITE
V04-000

EDTSLWRITE - write to a file
EDTSSEXI_CMD - EXIT line-mode command

16-Sep-1984 01:03:36
14-Sep-1984 12:23:47

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[EDIT.SRC]LWRITE.BLI;1

Page 22
(4)

788 1386 2 : RANGE [PREV_RANGE] = 0;
789 1387 2 :+ Position into the MAIN buffer.
790 1388 2 :-
791 1389 2 :+ EDT\$SFND_BUF (UPLIT (%STRING ('MAIN')), 4);
792 1390 2 :-
793 1391 2 :+ Attempt to write the file, and set the EXITED flag if it succeeds.
794 1392 2 :-
795 1393 2 :+ EDTSSG_EXITD = WRITE_FILE (RANGE, 1);
796 1394 2 :-
797 1395 1 END; ! of routine EDTSSEXI_CMD

				004D9		.BLKB 3	
		4E 49 41 4D	004DC	P.AAB:	.ASCII \MAIN\		:
					.EXTRN EDT\$SFND_BUF, EDTSSG_EXITD		
					.EXTRN EDTSSG_RCOV_MOD		
				0000 0000	.ENTRY EDT\$SEXI_CMD, Save nothing		1331
	5E 25	00000000G	20 00	C2 E8	SUBL2 #32 SP		
	6E	0B02	8F B0	00005 0000C	BLBS EDT\$SG_RCOV_MOD, 1\$		1379
			14	AE D4	MOVW #2818, RANGE		1384
				04 DD	CLRL RANGE+20		1386
				E3 AF	PUSHL #4		1390
		00000000G 00	02	9F 00016	PUSHAB P.AAB		
			01	FB 00019	CALLS #2, EDT\$SFND_BUF		1394
			04	DD 00020	PUSHL #1		
	FAFE	CF	AE 02	9F 00022	PUSHAB RANGE		
	00000000G	00	50	FB 00025	CALLS #2, WRITE FILE		
			04	DO 0002A	MOVL R0, EDTSSG_EXITD		
				00031 1\$:	RET		1395

: Routine Size: 50 bytes. Routine Base: _EDT\$CODE + 04E0

: 798 1396 1

ED
VO

00

800 1397 1 XSBTTL 'EDTSSWR_CMD - WRITE line-mode command'
801 1398 1
802 1399 1 GLOBAL ROUTINE EDTSSWR_CMD ! WRITE line-mode command
803 1400 1 : NOVALUE =
804 1401 1
805 1402 1 ++
806 1403 1 FUNCTIONAL DESCRIPTION:
807 1404 1
808 1405 1 Command processing routine for the write command. Look at the current range;
809 1406 1 if it is null, use the whole buffer. Then write the file.
810 1407 1
811 1408 1 FORMAL PARAMETERS:
812 1409 1
813 1410 1
814 1411 1
815 1412 1
816 1413 1
817 1414 1
818 1415 1
819 1416 1
820 1417 1
821 1418 1
822 1419 1
823 1420 1
824 1421 1
825 1422 1
826 1423 1
827 1424 1
828 1425 1
829 1426 1
830 1427 1
831 1428 1
832 1429 1 --
833 1430 1
834 1431 2 BEGIN
835 1432 2
836 1433 2
837 1434 2
838 1435 2
839 1436 2
840 1437 2
841 1438 2
842 1439 2
843 1440 2
844 1441 2
845 1442 2
846 1443 2
847 1444 2
848 1445 2
849 1446 2
850 1447 2
851 1448 2
852 1449 2
853 1450 2
854 1451 2
855 1452 2
856 1453 2
1454 2
1455 2
1456 2
1457 2
1458 2
1459 2
1460 2
1461 2
1462 2
1463 2
1464 2
1465 2
1466 2
1467 2
1468 2
1469 2
1470 2
1471 2
1472 2
1473 2
1474 2
1475 2
1476 2
1477 2
1478 2
1479 2
1480 2
1481 2
1482 2
1483 2
1484 2
1485 2
1486 2
1487 2
1488 2
1489 2
1490 2
1491 2
1492 2
1493 2
1494 2
1495 2
1496 2
1497 2
1498 2
1499 2
1500 2
1501 2
1502 2
1503 2
1504 2
1505 2
1506 2
1507 2
1508 2
1509 2
1510 2
1511 2
1512 2
1513 2
1514 2
1515 2
1516 2
1517 2
1518 2
1519 2
1520 2
1521 2
1522 2
1523 2
1524 2
1525 2
1526 2
1527 2
1528 2
1529 2
1530 2
1531 2
1532 2
1533 2
1534 2
1535 2
1536 2
1537 2
1538 2
1539 2
1540 2
1541 2
1542 2
1543 2
1544 2
1545 2
1546 2
1547 2
1548 2
1549 2
1550 2
1551 2
1552 2
1553 2
1554 2
1555 2
1556 2
1557 2
1558 2
1559 2
1560 2
1561 2
1562 2
1563 2
1564 2
1565 2
1566 2
1567 2
1568 2
1569 2
1570 2
1571 2
1572 2
1573 2
1574 2
1575 2
1576 2
1577 2
1578 2
1579 2
1580 2
1581 2
1582 2
1583 2
1584 2
1585 2
1586 2
1587 2
1588 2
1589 2
1590 2
1591 2
1592 2
1593 2
1594 2
1595 2
1596 2
1597 2
1598 2
1599 2
1600 2
1601 2
1602 2
1603 2
1604 2
1605 2
1606 2
1607 2
1608 2
1609 2
1610 2
1611 2
1612 2
1613 2
1614 2
1615 2
1616 2
1617 2
1618 2
1619 2
1620 2
1621 2
1622 2
1623 2
1624 2
1625 2
1626 2
1627 2
1628 2
1629 2
1630 2
1631 2
1632 2
1633 2
1634 2
1635 2
1636 2
1637 2
1638 2
1639 2
1640 2
1641 2
1642 2
1643 2
1644 2
1645 2
1646 2
1647 2
1648 2
1649 2
1650 2
1651 2
1652 2
1653 2
1654 2
1655 2
1656 2
1657 2
1658 2
1659 2
1660 2
1661 2
1662 2
1663 2
1664 2
1665 2
1666 2
1667 2
1668 2
1669 2
1670 2
1671 2
1672 2
1673 2
1674 2
1675 2
1676 2
1677 2
1678 2
1679 2
1680 2
1681 2
1682 2
1683 2
1684 2
1685 2
1686 2
1687 2
1688 2
1689 2
1690 2
1691 2
1692 2
1693 2
1694 2
1695 2
1696 2
1697 2
1698 2
1699 2
1700 2
1701 2
1702 2
1703 2
1704 2
1705 2
1706 2
1707 2
1708 2
1709 2
1710 2
1711 2
1712 2
1713 2
1714 2
1715 2
1716 2
1717 2
1718 2
1719 2
1720 2
1721 2
1722 2
1723 2
1724 2
1725 2
1726 2
1727 2
1728 2
1729 2
1730 2
1731 2
1732 2
1733 2
1734 2
1735 2
1736 2
1737 2
1738 2
1739 2
1740 2
1741 2
1742 2
1743 2
1744 2
1745 2
1746 2
1747 2
1748 2
1749 2
1750 2
1751 2
1752 2
1753 2
1754 2
1755 2
1756 2
1757 2
1758 2
1759 2
1760 2
1761 2
1762 2
1763 2
1764 2
1765 2
1766 2
1767 2
1768 2
1769 2
1770 2
1771 2
1772 2
1773 2
1774 2
1775 2
1776 2
1777 2
1778 2
1779 2
1780 2
1781 2
1782 2
1783 2
1784 2
1785 2
1786 2
1787 2
1788 2
1789 2
1790 2
1791 2
1792 2
1793 2
1794 2
1795 2
1796 2
1797 2
1798 2
1799 2
1800 2
1801 2
1802 2
1803 2
1804 2
1805 2
1806 2
1807 2
1808 2
1809 2
1810 2
1811 2
1812 2
1813 2
1814 2
1815 2
1816 2
1817 2
1818 2
1819 2
1820 2
1821 2
1822 2
1823 2
1824 2
1825 2
1826 2
1827 2
1828 2
1829 2
1830 2
1831 2
1832 2
1833 2
1834 2
1835 2
1836 2
1837 2
1838 2
1839 2
1840 2
1841 2
1842 2
1843 2
1844 2
1845 2
1846 2
1847 2
1848 2
1849 2
1850 2
1851 2
1852 2
1853 2
1854 2
1855 2
1856 2
1857 2
1858 2
1859 2
1860 2
1861 2
1862 2
1863 2
1864 2
1865 2
1866 2
1867 2
1868 2
1869 2
1870 2
1871 2
1872 2
1873 2
1874 2
1875 2
1876 2
1877 2
1878 2
1879 2
1880 2
1881 2
1882 2
1883 2
1884 2
1885 2
1886 2
1887 2
1888 2
1889 2
1890 2
1891 2
1892 2
1893 2
1894 2
1895 2
1896 2
1897 2
1898 2
1899 2
1900 2
1901 2
1902 2
1903 2
1904 2
1905 2
1906 2
1907 2
1908 2
1909 2
1910 2
1911 2
1912 2
1913 2
1914 2
1915 2
1916 2
1917 2
1918 2
1919 2
1920 2
1921 2
1922 2
1923 2
1924 2
1925 2
1926 2
1927 2
1928 2
1929 2
1930 2
1931 2
1932 2
1933 2
1934 2
1935 2
1936 2
1937 2
1938 2
1939 2
1940 2
1941 2
1942 2
1943 2
1944 2
1945 2
1946 2
1947 2
1948 2
1949 2
1950 2
1951 2
1952 2
1953 2
1954 2
1955 2
1956 2
1957 2
1958 2
1959 2
1960 2
1961 2
1962 2
1963 2
1964 2
1965 2
1966 2
1967 2
1968 2
1969 2
1970 2
1971 2
1972 2
1973 2
1974 2
1975 2
1976 2
1977 2
1978 2
1979 2
1980 2
1981 2
1982 2
1983 2
1984 2
1985 2
1986 2
1987 2
1988 2
1989 2
1990 2
1991 2
1992 2
1993 2
1994 2
1995 2
1996 2
1997 2
1998 2
1999 2
2000 2
2001 2
2002 2
2003 2
2004 2
2005 2
2006 2
2007 2
2008 2
2009 2
2010 2
2011 2
2012 2
2013 2
2014 2
2015 2
2016 2
2017 2
2018 2
2019 2
2020 2
2021 2
2022 2
2023 2
2024 2
2025 2
2026 2
2027 2
2028 2
2029 2
2030 2
2031 2
2032 2
2033 2
2034 2
2035 2
2036 2
2037 2
2038 2
2039 2
2040 2
2041 2
2042 2
2043 2
2044 2
2045 2
2046 2
2047 2
2048 2
2049 2
2050 2
2051 2
2052 2
2053 2
2054 2
2055 2
2056 2
2057 2
2058 2
2059 2
2060 2
2061 2
2062 2
2063 2
2064 2
2065 2
2066 2
2067 2
2068 2
2069 2
2070 2
2071 2
2072 2
2073 2
2074 2
2075 2
2076 2
2077 2
2078 2
2079 2
2080 2
2081 2
2082 2
2083 2
2084 2
2085 2
2086 2
2087 2
2088 2
2089 2
2090 2
2091 2
2092 2
2093 2
2094 2
2095 2
2096 2
2097 2
2098 2
2099 2
2100 2
2101 2
2102 2
2103 2
2104 2
2105 2
2106 2
2107 2
2108 2
2109 2
2110 2
2111 2
2112 2
2113 2
2114 2
2115 2
2116 2
2117 2
2118 2
2119 2
2120 2
2121 2
2122 2
2123 2
2124 2
2125 2
2126 2
2127 2
2128 2
2129 2
2130 2
2131 2
2132 2
2133 2
2134 2
2135 2
2136 2
2137 2
2138 2
2139 2
2140 2
2141 2
2142 2
2143 2
2144 2
2145 2
2146 2
2147 2
2148 2
2149 2
2150 2
2151 2
2152 2
2153 2
2154 2
2155 2
2156 2
2157 2
2158 2
2159 2
2160 2
2161 2
2162 2
2163 2
2164 2
2165 2
2166 2
2167 2
2168 2
2169 2
2170 2
2171 2
2172 2
2173 2
2174 2
2175 2
2176 2
2177 2
2178 2
2179 2
2180 2
2181 2
2182 2
2183 2
2184 2
2185 2
2186 2
2187 2
2188 2
2189 2
2190 2
2191 2
2192 2
2193 2
2194 2
2195 2
2196 2
2197 2
2198 2
2199 2
2200 2
2201 2
2202 2
2203 2
2204 2
2205 2
2206 2
2207 2
2208 2
2209 2
2210 2
2211 2
2212 2
2213 2
2214 2
2215 2
2216 2
2217 2
2218 2
2219 2
2220 2
2221 2
2222 2
2223 2
2224 2
2225 2
2226 2
2227 2
2228 2
2229 2
2230 2
2231 2
2232 2
2233 2
2234 2
2235 2
2236 2
2237 2
2238 2
2239 2
2240 2
2241 2
2242 2
2243 2
2244 2
2245 2
2246 2
2247 2
2248 2
2249 2
2250 2
2251 2
2252 2
2253 2
2254 2
2255 2
2256 2
2257 2
2258 2
2259 2
2260 2
2261 2
2262 2
2263 2
2264 2
2265 2
2266 2
2267 2
2268 2
2269 2
2270 2
2271 2
2272 2
2273 2
2274 2
2275 2
2276 2
2277 2
2278 2
2279 2
2280 2
2281 2
2282 2
2283 2
2284 2
2285 2
2286 2
2287 2
2288 2
2289 2
2290 2
2291 2
2292 2
2

EDT\$LWRITE
VO4-000

EDT\$LWRITE - write to a file
EDT\$SWR_CMD - WRITE line-mode command

L 2
16-Sep-1984 01:03:36 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:23:47 DISK\$VMSMASTER:[EDT.SRC]LWRITE.BLI;1 Page 24
(5)

: 857 1454 2 EDTSSA_CUR_BUF = .SAV_BUF;
: 858 1455 2 EDTSSCPY_MEM (POS_SIZE, EDT\$SZ_RNG_ORIGPOS, ! first get the buffer address
: 859 1456 2 EDT\$SRD_CURLN (); ! do the positioning
: 860 1457 1 END; ! of routine EDT\$SWR_CMD

.EXTRN EDT\$SRD_CURLN

		007C 00000	.ENTRY	EDT\$WR_CMD, Save R2,R3,R4,R5,R6	: 1399
	56 00000000G	00 9E 00002	MOVAB	EDTSSA_CUR_BUF, R6	
	52	66 D0 00009	MOVL	EDTSSA_CUR_BUF, SAV_BUF	: 1445
	50 00000000G	00 D0 0000C	MOVL	EDTSSA_EXE_CURCMD, R0	: 1446
	50 04	A0 D0 00013	MOVL	4(R0), RANGE	
		01 A0 95 00017	TSTB	1(RANGE)	: 1448
		04 12 0001A	BNEQ	1S	
01	A0	0B 90 0001C	MOVB	#11, 1(RANGE)	
		7E D4 00020	1\$: CLRL	-(SP)	: 1450
		50 DD 00022	PUSHL	RANGE	
FACD	CF	02 FB 00024	CALLS	#2, WRITE FILE	
	66	52 D0 00029	MOVL	SAV_BUF, EDTSSA_CUR_BUF	: 1454
	50	66 D0 0002C	MOVL	EDTSSA_CUR_BUF, R0	: 1455
60	00000000G	00 0E 28 0002F	MOVC3	#14, EDT\$SZ_RNG_ORIGPOS, (R0)	
	00000000G	00 00 FB 00037	CALLS	#0, EDT\$RD_CURLN	: 1456
		04 0003E	RET		: 1457

: Routine Size: 63 bytes, Routine Base: _EDT\$CODE + 0512

: 861 1458 1
: 862 1459 1 !<BLF/PAGE>

EDT\$LWRITE M 2
 V04-000 EDT\$LWRITE - write to a file 16-Sep-1984 01:03:36 VAX-11 Bliss-32 v4.0-742 Page 25
 EDTSSWR_CMD - WRITE line-mode command 14-Sep-1984 12:23:47 DISK\$VMSMASTER:[EDT.SRC]LWRITE.BLI:1 (6)
 :: 864 1460 1 END
 :: 865 1461 1
 :: 866 1462 0 ELUDOM ! of module EDT\$LWRITE

PSECT SUMMARY

Name	Bytes	Attributes
_EDT\$CODE	1361	NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)

Library Statistics

File	Total	Symbols	Pages	Processing Time
	Total	Loaded	Mapped	
\$255\$DUA28:[EDT.SRC]EDT.L32:1	377	109	40	00:00.2
\$255\$DUA28:[EDT.SRC]PSECTS.L32:1	2	1	7	00:00.1
\$255\$DUA28:[SYSLIB]STARLET.L32:1	9776	7	581	00:04.1
\$255\$DUA28:[EDT.SRC]SUPPORTS.L32:1	2	1	5	00:00.2

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACEBACK/LIS=LIS\$:\$LWRITE/OBJ=OBJ\$:\$LWRITE MSRC\$:\$LWRITE.BLI/UPDATE=(ENH\$:\$LWRITE)

Size: 1346 code + 15 data bytes
 Run Time: 01:00.3
 Elapsed Time: 01:15.8
 Lines/CPU Min: 1454
 Lexemes/CPU-Min: 7386
 Memory Used: 354 pages
 Compilation Complete

0136 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

LINCL
LIS

LTYPE
LIS

LQUERY
LIS

LSUB
LIS

LRES
LIS

LTADJ
LIS

LPRINT
LIS

GETSTR
LIS

LSUBSN
LIS

LPUTCHR
LIS

LSHOW
LIS

MOVE
LIS

LWRITE
LIS

LSET
LIS

LINSERT
LIS

LSUBS
LIS

LNONCTG
LIS

0137 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

LXCOMM00
LIS

MCGETLIN
LIS

MCRIGHT
LIS

MCHANGE
LIS

MACCAL
LIS

NOOPEN
LIS

PRAPPNUM
LIS

LXPRINT
LIS

MCTOP
LIS

PRGETTOK
LIS

PRISTOK
LIS

MCDOWN
LIS

MCLEFT
LIS

MCBOTTOM
LIS

PAUDIT
LIS

PRGETCHR
LIS

LXCOM
LIS

MAIN
LIS

MCLUP
LIS